



Managed App Configuration for App Developers

December 06, 2016

Copyright © 2016 MobileIron, Inc.

This work is licensed under a Creative Commons Attribution 4.0
License: <http://creativecommons.org/licenses/by/4.0>.

Contents

Chapter 1	Managed App Configuration Overview.....	6
	Benefits of managed app configuration	7
	Managed app configuration flow	8
	Configuration versioning	10
Chapter 2	Implementing Managed App Configuration	11
	Creating an XML document according to the schema	12
	Key-value pairs that your app expects to receive	12
	Default values for configuration settings	13
	Admin console user interface directives	13
	Localized strings for the admin console	14
	How an EMM service’s admin console chooses a localized string	15
	The version of the configuration settings	15
	Validating and submitting your XML document	17
	Modifying your app to consume the managed app configuration	18
	Consuming the configuration settings from NSUserDefaults	18
	Example of NSUserDefaults format	18
Chapter 3	Managed App Configuration XML Schema Reference	19
	Managed app configuration XML schema elements summary	20
	The root element managedApplicationConfiguration	22
	managedApplicationConfiguration	22
	Contains these elements	22
	Configuration settings version element	23
	version	23
	Parent element	23
	Bundle ID element	24
	bundleId	24
	Parent element	24
	The dictionary of key elements	25
	dict	25
	Parent element	25
	Contains these elements	25
	Key elements	26
	boolean	26
	Parent element	26
	Attribute	27
	Contains these elements	27
	Example	27
	date	27
	Parent element	27
	Attribute	28
	Contains these elements	28

Example	28
float	28
Parent element	28
Attribute	29
Contains these elements	29
Examples	29
floatArray	30
Parent element	30
Attribute	30
Contains these elements	30
Example	30
integer	31
Parent element	31
Attribute	31
Contains these elements	31
Examples	32
integerArray	32
Parent element	32
Attributes	33
Contains these elements	33
Example	33
string	33
Parent element	33
Attribute	34
Contains these elements	34
Example	34
stringArray	34
Parent element	34
Attribute	35
Contains these elements	35
Example	35
Elements for default values and constraints for keys	36
constraint	36
Parent elements	36
Attributes	37
Contains these elements	38
Example for an integer or integerArray element	38
Example for a string or stringArray element	38
defaultValue	38
Parent elements	39
Contains these elements	39
Examples for a string element	40
Example for a stringArray element	40
Example for an integer element	40
Example for a date element	40
deviceVariable	40
Parent element	40
Attribute	41
Example	41
userVariable	41
Parent element	41
Attribute	42
Example	42
value	42
Parent elements	42
Example	42
values	42

Parent element	43
Contains these elements	43
Example	43
Presentation-related elements	44
description	44
Parent element	44
Contains these elements	44
Example	44
field	45
Parent elements	45
Attributes	46
Contains these elements	46
fieldGroup	47
Parent element	47
Contains these elements	47
Example	47
label	47
Parent element	48
Contains these elements	48
Example	48
language	48
Parent elements	48
Attribute	48
Example	48
name	48
Parent element	49
Contains these elements	49
Example	49
option	49
Parent element	49
Attributes	50
Contains these elements	50
Example	50
options	50
Parent element	50
Contains these elements	51
Example	51
presentation	51
Parent element	51
Attribute	52
Contains these elements	52
Example	52
EMM service user variables	53
EMM service device variables	54
Managed app configuration XML document example	55
plist format sent to device	58
plist format	58
plist example with each type of key	59

Chapter 1

Managed App Configuration Overview

Enterprises use Enterprise Mobility Management (EMM) services to manage mobile devices, including the apps and content on those devices. Managed app configuration streamlines the purchasing, discovery, deployment, configuration, feedback and adoption of your apps by enterprise customers.

Managed app configuration uses native iOS capabilities for managed apps. Managed apps are apps that are distributed and managed using an EMM service.

An app typically requires some configuration on the device, such as:

- user information
- server information
- whether particular features should be enabled

Managed app configuration:

- Allows your app to automatically receive default configuration settings from the EMM service so that device users can start using the app immediately.
- Provides enterprise administrators an easy graphical user interface to configure your app to meet their needs, if default configuration settings are not sufficient.
- Allows your app to receive a configuration setting that specifies the version of the settings received from the EMM service. The version setting allows each version of the app to correctly handle different versions of configuration settings.

Important: iOS stores the configuration settings unencrypted on the device. Therefore, as a best practice, do not use managed app configuration for sensitive information such as passwords or private keys.

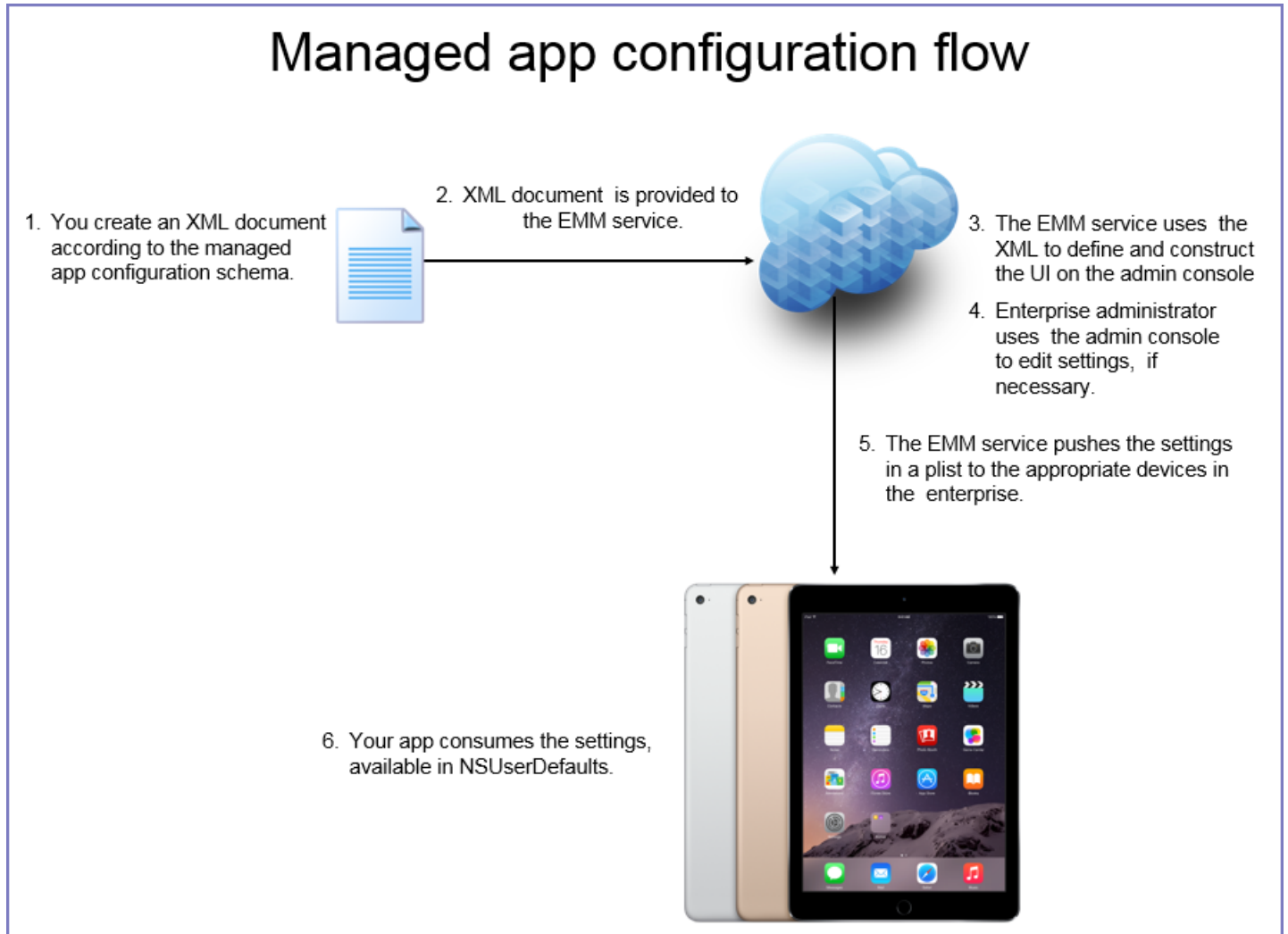
Benefits of managed app configuration

Managed app configuration results in increased adoption of your apps because:

- When all configurable settings have a default value, your app is ready to use without either the enterprise administrator or device user doing any work.
- The EMM service administrators follow an easy to use interface to configure the app if they want to change default values or if some values have no defaults. Using a straightforward user interface avoids configuration mistakes and encourages further distribution to device users.
- Enterprise device users do not have to manually configure the app, thereby avoiding manual configuration mistakes that result in help desk calls. Eliminating manual configuration also provides a better user experience, which results in more positive reviews of your apps. Moreover, device users will not give avoidable negative reviews that can result from configuration mistakes.

Managed app configuration flow

Managed app configuration works as follows:



1. You provide the app's configurable settings as an XML document, according to the managed app configuration schema.
The schema allows you to specify setting names, types, and possible values. The type can be an input field, a select field such as a drop-down list, or a check box.
You also specify default values for each setting, if appropriate.
2. The XML document is provided to the EMM service.
3. The EMM service uses your XML document to determine how to display an easy to use interface for the configurable settings on its admin console.
4. The enterprise administrator modifies the default values, if necessary, and enters values for configurable settings with no defaults, if any.
5. The EMM service pushes the settings in a plist to the devices that are using your app.
iOS puts the settings in the plist into NSUserDefaults.

6. Your app gets the settings from `NSUserDefaults`, and then stores the settings as needed and takes the appropriate actions.
For example, your app connects with a specified service with a specified user ID, or restricts or enables access to certain app features.

Configuration versioning

The managed app configuration schema allows for versioning your app's configurable settings. As your app evolves from version to version, the configuration requirements can change. For example:

- You add possible values to existing settings.
- You add new settings.
- You remove settings.

By specifying the version of the configuration settings in the XML document that you provide to the EMM service, your app will receive the version in the configuration settings sent from the EMM service. Then, your app can compare the version in the configuration settings with the version that it is using and take the appropriate actions.

Chapter 2

Implementing Managed App Configuration

Implementing managed app configuration with your app involves the following tasks:

1. [“Creating an XML document according to the schema” on page 12](#)
2. [“Validating and submitting your XML document” on page 17](#)
3. [“Modifying your app to consume the managed app configuration” on page 18](#)

Creating an XML document according to the schema

Create an XML document according to the managed app configuration schema. You can use the [“Managed app configuration XML document example” on page 55](#) as a starting point.

In the XML document, specify the following:

1. [“Key-value pairs that your app expects to receive” on page 12](#)
2. [“Default values for configuration settings” on page 13](#)
3. [“Admin console user interface directives” on page 13](#)
4. [“Localized strings for the admin console” on page 14](#)
5. [“The version of the configuration settings” on page 15](#)

Important: After you have created the XML document, validate its content. See [“Validating and submitting your XML document” on page 17](#).

Details about the XML elements are in [“Managed App Configuration XML Schema Reference” on page 19](#).

Key-value pairs that your app expects to receive

You specify in the XML document a key for each configuration setting that your app expects to receive. The EMM service delivers the keys and their default values or administrator-assigned values to the device for your app to consume. The key-value pairs are available to your app in `NSUserDefaults`.

Use these key elements to specify the key-value pairs:

- `boolean`
- `date`
- `float`
- `floatArray`
- `integer`
- `integerArray`
- `string`
- `stringArray`

For each key-value pair that you specify, define the following:

- constraints on allowed values, such as a minimum and maximum value.
You specify the constraints in a `constraint` element contained in a key element.
- the default value to display on the admin console and to deliver to your app if the enterprise administrator makes no changes.

You specify the default value in a `defaultValue` element contained in a key element.

Important: Carefully consider the best default value for each key. When enterprise administrators do not need to make any changes on the EMM service admin console, your app will be ready to use without either the enterprise administrator or device user doing any work.

Default values for configuration settings

You specify in the XML document the default value for each configuration setting. When the EMM service admin console displays a configuration setting field, it uses the default value as the initial setting. If the enterprise administrator makes no changes, the EMM service delivers the default value in the key-value pair.

Specify the default value in the [defaultValue](#) element of key element. The default value can be either:

- a literal, such as 10 or "USD"
- an EMM service variable for [string](#) or [stringArray](#) key elements.

EMM service variables are either:

- device variables, such as `phoneNumber` OR `serialNumber`
- user variables, such as `username` OR `emailAddress`

Note: Some of the user variables depend on integration with either Active Directory or other LDAP server.

The following [string](#) element shows an example of using an EMM service variable as the default value:

```
<string keyName="mobileNumber">
  <defaultValue>
    <deviceVariable value="phoneNumber"/>
  </defaultValue>
  <constraint nullable="true" min="0" max="255"/>
</string>
```

In this example:

- The admin console displays the EMM service's name for the variable in the field corresponding to this key. For example, the admin console displays `$(phoneNumber)`.
- When the EMM service delivers the key-value pair to your app, it delivers the value for that device's user. For example, it delivers the value 888-555-1212.

For the list of available variables, see:

- ["EMM service user variables" on page 53](#)
- ["EMM service device variables" on page 54](#)

Admin console user interface directives

You specify in the XML document how you want the EMM service to display your app's configuration settings on its admin console. Specify this information in the [presentation](#) element.

For each configuration setting, indicate:

- the kind of user interface control to use
- whether it is part of a group of controls that are displayed together in a group box
- the label and description to display

For "select" controls where you choose an option (for example, a drop-down list), indicate:

- the name of each option to display
- which option or options to initially select

- the value to send to the app when each option is selected

If you do not provide user interface directives in the XML document, the EMM service uses information from the elements describing each key-value pair to determine how to create a corresponding UI control. However, it is recommended that you always provide this information for the best possible enterprise administrator experience, resulting in the most successful deployment of your app.

Localized strings for the admin console

You provide localized strings in the XML document. The localized strings are the descriptive text that appears on the admin console for your configuration settings. The localized strings ensure that the enterprise administrator understands the purpose of each configurable setting for your app.

You specify the localized strings in [language](#) elements. Each language element contains one localized string for one language or language-region code, such as en or en-US. For example:

```
<language value="en-US">Start Time</language>
<language value="es-ES">Hora de Inicio</language>
```

The following elements contain language elements:

- [label](#) elements, which specify the label of the UI control for a configuration setting on the admin console
- [description](#) elements, which provide descriptive text about the UI control
- [option](#) elements, which specify the name of an option for a `select` or `multiselect` UI control
- [name](#) elements, which specify the name of a group box

The EMM service determines which of the localized strings to use. It looks for a [language](#) element in the following order, and if found, uses its localized string:

1. The primary language, such as the language that the browser running the admin console is using. The primary language depends on the implementation of the EMM service.
2. The language specified by the [presentation](#) element's `defaultLocale` attribute.

Note: Make sure you provide language elements with `value` attributes that *exactly* match the `defaultLocale` attribute.

The following table shows what the admin console displays if the EMM service does not find either of the above languages among the [language](#) elements:

Element containing language element	What the admin console displays if no matching language element is provided
label element	The <code>keyName</code> attribute of the associated key element.
description element	No description is displayed.
option element	The EMM service determines how to display each option in the <code>select</code> or <code>multiselect</code> UI control. For example, the EMM service can display the value for the option. The value is specified in the values elements of the constraint element for the key element corresponding to the field.
name element	No name is displayed for the group box.

How an EMM service's admin console chooses a localized string

An EMM service displays its admin console using a language-region, such as en-US or de. How the EMM service chooses that language can vary. Also, the set of supported languages vary by EMM service. Contact the EMM provider to find out its supported languages.

Given the display language, the EMM service compares the display language with the provided language elements in the XML document to determine which localized string to use. For example, the admin console might choose language elements in the following order:

1. a language element with the same language and region as the admin console
2. a language element with just the same language, but a different region or no specified region
3. a language element corresponding to the `defaultLocale` attribute in the [presentation](#) element

The following table shows more specifically how an EMM service's admin console might choose the language element, although this can vary among EMM providers:

Admin console language-region setting	Example of order that an admin console might look for a language element in the XML document
xx-YY For example: en-US	<ol style="list-style-type: none">1. A language element with the <code>value</code> attribute xx-YY. That is, the attribute exactly matches the admin console's language and region.2. A language element with a <code>value</code> attribute that begins with xx. The region portion of the <code>value</code> attribute, if included, does not have to match YY. That is, the attribute matches just the admin console's language, but not the region. For example, consider an XML document that has en-US strings but no en-GB strings. An admin console for which the language setting is en_GB uses the en-US strings from the XML document.3. A language element with a <code>value</code> attribute that exactly matches the presentation element's <code>defaultLocale</code> attribute.
xx For example: es	<ol style="list-style-type: none">1. A language element with a <code>value</code> attribute that begins with xx. That is, the attribute matches the admin console's language, and the attribute's region is ignored.2. A language element with a <code>value</code> attribute that exactly matches the presentation element's <code>defaultLocale</code> attribute.

The version of the configuration settings

You specify in the XML document the version of the configuration settings in the [version](#) element. The EMM service delivers this version number as a key-value pair along with the configuration settings. When the app receives the configuration settings, it can compare the version received with the version it uses.

Note: The delivered key in `NSUserDefaults` is named `ManagedAppConfigurationVersion`.

Consider how you want the app to handle older and newer versions of configurable settings. Ideally, a new version of the app still works with an older version's configuration settings, and an older version of the app continues to work with a newer version's configuration settings.

The cases to handle include when a newer version of the app:

- Added, removed, or changed possible values to existing settings.
- Added settings.
- Removed settings.

Validating and submitting your XML document

Validate your XML document against the managed app configuration schema XSD file. Various web tools are available for validating an XML document against an XSD file. If you make changes to the XML document, be sure to validate it again.

Once you have validated your final XML document, submit it to the EMM provider.

Modifying your app to consume the managed app configuration

The EMM service pushes your app's configuration settings to the devices that are using your app. An app accesses the delivered settings by using the iOS managed app configuration capability. The delivered settings, and changes to the settings, are available in `NSUserDefaults`.

Details about how to access `NSUserDefaults` are described by Apple at

<https://developer.apple.com/library/ios/samplecode/sc2279/Introduction/Intro.html>.

Consuming the configuration settings from `NSUserDefaults`

At a high-level, do the following tasks in your app:

1. Add a Notification Center observer to be alerted of any change to `NSUserDefaults`.
2. Get the shared `NSUserDefaults` object to initially get the configuration settings.
3. Get the `NSDictionary` object for the key `@“com.apple.configuration.managed”` from the shared `NSUserDefaults` object.
4. Extract the `ManagedAppConfigurationVersion` key-value pair from the `NSDictionary` object.
Note: The name of the key in `NSUserDefaults` is `ManagedAppConfigurationVersion`, but in the XML document, you specify the version in a key named `version`.
5. Extract configuration settings key-value pairs from the `NSDictionary` object.
The `NSDictionary` object contains a key-value pair for each key element (`string`, `stringArray`, `integer`, and so on) that you specified in the XML document. The name of the key is the `keyName` attribute of the key element. The type of value depends on the type of the key element.
Note: In the XML document, you specify key-value pairs for configuration settings within the `dict` element, but the `NSDictionary` in `NSUserDefaults` does not contain the `dict` element. The key-value pairs are contained in the `NSDictionary` object without an intervening layer.
6. Apply the configuration settings as appropriate to your app.
Always compare the value of the `ManagedAppConfigurationVersion` key with the configuration settings version that the app uses as part of determining how to apply the settings.
7. When the app receives notifications of changes, apply the updated configuration settings as appropriate to your app.

Example of `NSUserDefaults` format

Your app gets the `NSDictionary` object for the key `@“com.apple.configuration.managed”` from the shared `NSUserDefaults` object. The following JavaScript Object Notation (JSON) gives an example of the key-value pairs in the retrieved `NSDictionary` object:

```
{
  "ManagedAppConfigurationVersion": 2,
  "userDisplayName": "Jane Doe",
  "approvalAllowed": true,
  "serverURL": "myApp.EnterpriseServer.com"
}
```

Chapter 3

Managed App Configuration XML Schema Reference

- “Managed app configuration XML schema elements summary” on page 20
- “The root element managedApplicationConfiguration” on page 22
- “Configuration settings version element” on page 23
- “Bundle ID element” on page 24
- “The dictionary of key elements” on page 25
- “Key elements” on page 26
- “Elements for default values and constraints for keys” on page 36
- “Presentation-related elements” on page 44
- “EMM service user variables” on page 53
- “EMM service device variables” on page 54
- “Managed app configuration XML document example” on page 55
- “plist format sent to device” on page 58

Managed app configuration XML schema elements summary

Define your app's configuration requirements in an XML document. The overall structure of an XML document is:

managedApplicationConfiguration

<code>version (1)</code>	
<code>bundleId (1)</code>	
<code>dict (1)</code>	
<code>string (0+)</code>	Attribute: keyName
<code>defaultValue (0-1)</code>	
<code>value (0-1)</code>	
<code>userVariable (0-1)</code>	Attribute: value
<code>deviceVariable (0-1)</code>	Attribute: value
<code>constraint (0-1)</code>	Attributes: nullable, min, max, pattern
<code>values (0-1)</code>	
<code>value (1+)</code>	
<code>stringArray (0+)</code>	Attribute: keyName
<code>defaultValue (0-1)</code>	
<code>value (0+)</code>	
<code>userVariable (0+)</code>	Attribute: value
<code>deviceVariable (0+)</code>	Attribute: value
<code>constraint (0-1)</code>	Attributes: nullable, min, max, pattern
<code>values (0-1)</code>	
<code>value (1+)</code>	
<code>integer (0+)</code>	Attribute: keyName
<code>defaultValue (0-1)</code>	
<code>value (1)</code>	
<code>constraint (0-1)</code>	Attributes: nullable, min, max
<code>values (0-1)</code>	
<code>value (1+)</code>	
<code>integerArray (0+)</code>	Attribute: keyName
<code>defaultValue (0-1)</code>	
<code>value (0+)</code>	
<code>constraint (0-1)</code>	Attributes: nullable, min, max
<code>values (0-1)</code>	
<code>value (1+)</code>	
<code>float (0+)</code>	Attribute: keyName. Similar child elements as integer
<code>floatArray (0+)</code>	Attribute: keyName. Similar child elements as integerArray
<code>date (0+)</code>	Attribute: keyName
<code>defaultValue (0-1)</code>	
<code>value (1)</code>	
<code>constraint (0-1)</code>	Attributes: nullable, min, max
<code>boolean (0+)</code>	Attribute: keyName
<code>defaultValue (0-1)</code>	
<code>value (1)</code>	
<code>constraint (0-1)</code>	Attribute: nullable
<code>presentation (0-1)</code>	Attribute: defaultLocale
<code>fieldGroup (0+)</code>	
<code>name (1)</code>	
<code>language (1+)</code>	Attribute: value
<code>field (1+)</code>	
<code>field (0+)</code>	Attributes: keyName, type
<code>label (0-1)</code>	
<code>language (1+)</code>	Attribute: value
<code>description (0-1)</code>	
<code>language (1+)</code>	Attribute: value
<code>options (0-1)</code>	
<code>option (1+)</code>	Attributes: selected, value
<code>language (1+)</code>	Attribute: value

The following table describes the above notation for the number of occurrences of an element in its containing element:

Occurrence notation	Description
(1)	Exactly one instance of the element is required.
(1+)	One or more instances of the element are required.
(0-1)	The element is optional, but at most one instance is allowed.
(0+)	The element is optional, but multiple instances are allowed.

Details about each element follow.

See also:

- ["Creating an XML document according to the schema" on page 12](#)
- ["Managed app configuration XML document example" on page 55](#)

The root element managedApplicationConfiguration

managedApplicationConfiguration

managedApplicationConfiguration is the root element of the XML document.

Contains these elements

The managedApplicationConfiguration elements contains these elements in the listed order:

Element	Occurrences	Description
version	Required (1)	The version of the configuration settings that this XML document is associated with. The version is an integer.
bundleId	Required (1)	The bundle ID of the app. The bundleId element is a string.
dict	Required (1)	A list of key elements. Each key element defines a configuration setting that you require the EMM service to send to the app.
presentation	Optional (0 - 1)	Defines how you want each configuration setting to display on the EMM service admin console. If you do not provide a presentation element, the EMM service uses information from each key element to determine how to create a corresponding UI control.

Configuration settings version element

version

The `version` element specifies the version of the configuration settings that the XML document is associated with.

The `version` is an integer, 1 to 5 digits long.

Note: Although in the XML document, the key is named `version`, in `NSUserDefaults` the name of the key is `ManagedAppConfigurationVersion`.

Parent element

`managedApplicationConfiguration` (the root element)

Example

```
<version>2</version>
```

Bundle ID element

bundleId

The bundleId element specifies the bundle ID of the iOS app that the XML document is associated with.

The bundleId is a string.

Parent element

[managedApplicationConfiguration](#) (the root element)

Example

```
<bundleId>com.myCompanyName.mAppName</bundleId>
```


The dictionary of key elements

dict

The dict element is a list of key elements, where each key element defines a configuration setting that you require the EMM service to send to the app.

Note: In the XML document, you specify key-value pairs for configuration settings within the [dict](#) element. However, when the EMM service delivers the configuration settings to the device, the `NSDictionary` in `NSUserDefaults` does not contain the dict element. The key-value pairs are contained in the `NSDictionary` object without an intervening layer.

Parent element

[managedApplicationConfiguration](#) (the root element)

Contains these elements

The dict element must have at least one of the following child elements. Each child element defines a configuration setting that you require the EMM service to send to the app in a key-value pair:.

Element	Occurrences	Description
boolean	Optional (0 or more)	Defines a configuration setting that is a boolean.
date	Optional (0 or more)	Defines a configuration setting that is a date.
float	Optional (0 or more)	Defines a configuration setting that is a float.
floatArray	Optional (0 or more)	Defines a configuration setting that is a float array.
integer	Optional (0 or more)	Defines a configuration setting that is an integer.
integerArray	Optional (0 or more)	Defines a configuration setting that is an integer array.
string	Optional (0 or more)	Defines a configuration setting that is a string.
stringArray	Optional (0 or more)	Defines a configuration setting that is a string array.

Key elements

The key elements, in alphabetical order, are:

- [boolean](#)
- [date](#)
- [float](#)
- [floatArray](#)
- [integer](#)
- [integerArray](#)
- [string](#)
- [stringArray](#)

boolean

The boolean element defines a key that you require the EMM service to send to the app in a key-value pair. The value must be a boolean.

Parent element

[dict](#)

Attribute

Attribute	Required or optional	Description
keyName	Required	<p>The name of the key in the key-value pair that the EMM service sends to the app.</p> <p>The app uses this name to look up the configuration setting in the managed app configuration dictionary (NSDictionary).</p> <p>The name of the key must:</p> <ul style="list-style-type: none">• Contain only alphanumeric characters, spaces, _ (underscore), and - (dash).• Be unique within the dict element. <p>Example:</p> <pre>keyName="allowAdvancedOptions"</pre>

Contains these elements

The boolean element contains these elements in the listed order.

Element	Occurrences	Description
defaultValue	Optional (0 or 1)	<p>The default value for the EMM service to send to the app for the boolean, and to display in the corresponding user interface control.</p> <p>The default value must be a value element containing either <code>true</code> or <code>false</code>.</p>
constraint	Optional (0 or 1)	<p>The validation rules for the value of the configuration setting.</p>

Example

```
<boolean keyName="advancedOptionsAllowed">  
  <defaultValue>  
    <value>true</value>  
  </defaultValue>  
</boolean>
```

date

The date element defines a key that you require the EMM service to send to the app in a key-value pair. The value must be a date in [ISO-8601](#) format.

Examples of ISO-8601 format are:

```
2015-07-25  
2015-07-25T21:19Z
```

Parent element

[dict](#)

Attribute

Attribute	Required or optional	Description
keyName	Required	<p>The name of the key in the key-value pair that the EMM service sends to the app.</p> <p>The app uses this name to look up the configuration setting in the managed app configuration dictionary (NSDictionary).</p> <p>The name of the key must:</p> <ul style="list-style-type: none">• Contain only alphanumeric characters, spaces, _ (underscore), and - (dash).• Be unique within the dict element. <p>Example:</p> <pre>keyName="expiration"</pre>

Contains these elements

The date element contains these elements in the listed order.

Element	Occurrences	Description
defaultValue	Optional (0 or 1)	<p>The default value for the EMM service to send to the app for the date, and to display in the corresponding user interface control.</p> <p>The default value must be a value element containing a date in ISO-8601 format.</p>
constraint	Optional (0 or 1)	<p>The validation rules for the value of the configuration setting.</p>

Example

```
<date keyName="startDate">  
  <defaultValue>  
    <value>2015-07-25T21:19:30Z</value>  
  </defaultValue>  
</date>
```

float

The float element defines a key that you require the EMM service to send to the app in a key-value pair. The value must be a float.

Parent element

[dict](#)

Attribute

Attribute	Required or optional	Description
keyName	Required	<p>The name of the key in the key-value pair that the EMM service sends to the app.</p> <p>The app uses this name to look up the configuration setting in the managed app configuration dictionary (NSDictionary).</p> <p>The name of the key must:</p> <ul style="list-style-type: none"> • Contain only alphanumeric characters, spaces, _ (underscore), and - (dash). • Be unique within the dict element. <p>Example: keyName="price"</p>

Contains these elements

The float element contains these elements in the listed order.

Element	Occurrences	Description
defaultValue	Optional (0 or 1)	<p>The default value for the EMM service to send to the app for the float, and to display in the corresponding user interface control.</p> <p>The default value must be a value element containing a float.</p>
constraint	Optional (0 or 1)	The validation rules for the value of the configuration setting.

Examples

```
<float keyName="floatkey1">
  <defaultValue>
    <value>1.50</value>
  </defaultValue>
  <constraint min="1" max="2"/>
</float>
```

```
<float keyName="floatkey2">
  <defaultValue>
    <value>20.5</value>
  </defaultValue>
  <constraint>
    <values>
      <value>20.0</value>
      <value>20.1</value>
      <value>20.2</value>
      <value>20.3</value>
      <value>20.4</value>
      <value>20.5</value>
    </values>
  </constraint>
</float>
```

floatArray

The floatArray element defines a key that you require the EMM service to send to the app in a key-value pair. The value must be a float array.

Parent element

[dict](#)

Attribute

Attribute	Required or optional	Description
keyName	Required	<p>The name of the key in the key-value pair that the EMM service sends to the app.</p> <p>The app uses this name to look up the configuration setting in the managed app configuration dictionary (NSDictionary).</p> <p>The name of the key must:</p> <ul style="list-style-type: none">• Contain only alphanumeric characters, spaces, _ (underscore), and - (dash).• Be unique within the dict element. <p>Example: keyName="prices"</p>

Contains these elements

The floatArray element contains these elements in the listed order.

Element	Occurrences	Description
defaultValue	Optional (0 or 1)	<p>The default values for the EMM service to send to the app for the float array, and to display in the corresponding user interface control.</p> <p>Each default value must be a value element containing a float.</p>
constraint	Optional (0 or 1)	<p>The validation rules for the value of the configuration setting.</p>

Example

```
<floatArray keyName="floatarraykey1">
  <defaultValue>
    <value>1.5</value>
  </defaultValue>
  <constraint>
    <values>
      <value>1.3</value>
      <value>1.4</value>
      <value>1.5</value>
    </values>
  </constraint>
</floatArray>
```

integer

The integer element defines a key that you require the EMM service to send to the app in a key-value pair. The value must be an integer.

Parent element

[dict](#)

Attribute

Attribute	Required or optional	Description
keyName	Required	<p>The name of the key in the key-value pair that the EMM service sends to the app.</p> <p>The app uses this name to look up the configuration setting in the managed app configuration dictionary (NSDictionary).</p> <p>The name of the key must:</p> <ul style="list-style-type: none">• Contain only alphanumeric characters, spaces, _ (underscore), and - (dash).• Be unique within the dict element. <p>Example:</p> <pre>keyName="size"</pre>

Contains these elements

The integer element contains these elements in the listed order.

Element	Occurrences	Description
defaultValue	Optional (0 or 1)	<p>The default value for the EMM service to send to the app for the integer, and to display in the corresponding user interface control.</p> <p>The default value must be a value element containing an integer.</p>
constraint	Optional (0 or 1)	<p>The validation rules for the value of the configuration setting.</p>

Examples

```
<integer keyName="percent">
  <defaultValue>
    <value>20</value>
  </defaultValue>
  <constraint min="1" max="100" nullable="true"/>
</integer>
```

```
<integer keyName="speed">
  <defaultValue>
    <value>25</value>
  </defaultValue>
  <constraint>
    <values>
      <value>15</value>
      <value>20</value>
      <value>25</value>
      <value>30</value>
    </values>
  </constraint>
</integer>
```

integerArray

The integerArray element defines a key that you require the EMM service to send to the app in a key-value pair. The value must be an integer array.

Parent element

[dict](#)

Attributes

Attribute	Required or optional	Description
keyName	Required	<p>The name of the key in the key-value pair that the EMM service sends to the app.</p> <p>The app uses this name to look up the configuration setting in the managed app configuration dictionary (NSDictionary).</p> <p>The name of the key must:</p> <ul style="list-style-type: none">• Contain only alphanumeric characters, spaces, _ (underscore), and - (dash).• Be unique within the dict element. <p>Example: keyName="ratings"</p>

Contains these elements

The integer array element contains these elements in the listed order.

Element	Occurrences	Description
defaultValue	Optional (0 or 1)	<p>The default values for the EMM service to send to the app for the integer array, and to display in the corresponding user interface control.</p> <p>Each default value must be a value element containing an integer.</p>
constraint	Optional (0 or 1)	<p>The validation rules for the value of the configuration setting.</p>

Example

```
<integerArray keyName="size">
  <defaultValue>
    <value>2048</value>
  </defaultValue>
  <constraint>
    <values>
      <value>1024</value>
      <value>2048</value>
      <value>4096</value>
    </values>
  </constraint>
</integerArray>
```

string

The string element defines a key that you require the EMM service to send to the app in a key-value pair. The value must be a string.

Parent element

[dict](#)

Attribute

Attribute	Required or optional	Description
keyName	Required	<p>The name of the key in the key-value pair that the EMM service sends to the app.</p> <p>The app uses this name to look up the configuration setting in the managed app configuration dictionary (NSDictionary).</p> <p>The name of the key must:</p> <ul style="list-style-type: none">• Contain only alphanumeric characters, spaces, _ (underscore), and - (dash).• Be unique within the dict element. <p>Example:</p> <pre>keyName="zipcode"</pre>

Contains these elements

The string element contains these elements in the listed order.

Element	Occurrences	Description
defaultValue	Optional (0 or 1)	<p>The default value for the EMM service to send to the app for the string, and to display in the corresponding user interface control.</p> <p>The default value can be one of the following:</p> <ul style="list-style-type: none">• a string literal in a value element• an EMM service variable in a userVariable or deviceVariable element. <p>Note: The defaultValue is not localizable.</p>
constraint	Optional (0 or 1)	The validation rules for the value of the configuration setting.

Example

```
<string keyName="commonName">  
  <defaultValue>  
    <userVariable value="cn"/>  
  </defaultValue>  
  <constraint nullable="true" pattern="CN=.*"/>  
</string>
```

stringArray

The stringArray element defines a key that you require the EMM service to send to the app in a key-value pair. The value must be a string array.

Parent element

[dict](#)

Attribute

Attribute	Required or optional	Description
keyName	Required	<p>The name of the key in the key-value pair that the EMM service sends to the app.</p> <p>The app uses this name to look up the configuration setting in the managed app configuration dictionary (NSDictionary).</p> <p>The name of the key must:</p> <ul style="list-style-type: none">• Contain only alphanumeric characters, spaces, _ (underscore), and - (dash).• Be unique within the dict element. <p>Example:</p> <pre>keyName="countries"</pre>

Contains these elements

The `stringArray` element contains these elements in the listed order.

Element	Occurrences	Description
defaultValue	Optional (0 or 1)	<p>The default values for the EMM service to send to the app for the string array, and to display in the corresponding user interface control.</p> <p>Each default value can be one of the following:</p> <ul style="list-style-type: none">• a string literal in a value element• an EMM service variable in a userVariable or deviceVariable element. <p>Note: The <code>defaultValue</code> is not localizable.</p>
constraint	Optional (0 or 1)	The validation rules for the value of the configuration setting.

Example

```
<stringArray keyName="userEmailAddress">  
  <defaultValue>  
    <userVariable value="emailAddress"/>  
  </defaultValue>  
  <constraint min="1" max="255" pattern="[A-Za-z]@company.com"/>  
</stringArray>
```

Elements for default values and constraints for keys

The default values for key elements are specified by the following elements:

```
defaultValue (0-1)
  value (0-1)
  userVariable (0-1)  Attribute: value
  deviceVariable (0-1) Attribute: value
```

Constraints for the values of key elements are specified by the following elements:

```
constraint (0-1)      Attributes: nullable, min, max, pattern
  values (0-1)
    value (1+)
```

These elements are listed below in alphabetical order.

constraint

The constraint element defines the validation rules for the value (or values) of the configuration setting. The EMM service uses these rules to make sure values that the enterprise administrator enters on the admin console are valid.

Parent elements

- [boolean](#)
- [date](#)
- [float](#)
- [floatArray](#)
- [integer](#)
- [integerArray](#)
- [string](#)
- [stringArray](#)

Attributes

The constraint element can have the following attributes, all of which are optional:

Attribute	Description
nullable	<p>Set <code>nullable</code> to <code>true</code> if you want to allow the value in the key-value pair to be null. The default value is <code>false</code>.</p> <p>Example: <code>nullable="true"</code></p>
min	<p>The lower bound for the value. If the <code>min</code> attribute is not specified, the value has no lower bound.</p> <p>The meaning depends on the parent element of the constraint element.</p> <ul style="list-style-type: none">• <code>boolean</code> - not applicable• <code>date</code> - the minimum date• <code>float</code> - the minimum float value• <code>floatArray</code> - the minimum number of elements in the float array• <code>integer</code> - the minimum integer value• <code>integerArray</code> - the minimum number of elements in the integer array• <code>string</code> - the minimum number of characters in the string• <code>stringArray</code> - the minimum number of elements in the string array <p>Example: <code>min="10"</code></p>
max	<p>The upper bound for the value. If the <code>max</code> attribute is not specified, the value has no upper bound.</p> <p>The meaning depends on the parent element of the constraint element.</p> <ul style="list-style-type: none">• <code>boolean</code> - not applicable• <code>date</code> - the maximum date• <code>float</code> - the maximum float value• <code>floatArray</code> - the maximum number of elements in the float array• <code>integer</code> - the maximum integer value• <code>integerArray</code> - the maximum number of elements in the integer array• <code>string</code> - the maximum number of characters in the string• <code>stringArray</code> - the maximum number of elements in the string array <p>Example: <code>max="100"</code></p>
pattern	<p>A regular expression to use to validate a string value of a key.</p> <p>This attribute is allowed only when the parent element of the constraint element is <code>string</code> or <code>stringArray</code>.</p> <p>If <code>pattern</code> is specified, the <code>min</code> and <code>max</code> attributes are ignored.</p> <p>Example: <code>pattern="[A-Za-z]@company.com"</code></p>

Contains these elements

Element	Occurrences	Description
values	Optional (0 or 1)	<p>A set of value elements, which define the set of valid values of a float, floatArray, integer, integerArray, string, or stringArray, element.</p> <p>Use the values element when you want to give the enterprise administrator a list of valid values to choose from.</p> <p>This element is not allowed when the parent element of the constraint element is boolean or date.</p>

Example for an integer or integerArray element

Using a minimum and maximum value:

```
<constraint nullable="true" min="0" max="10"/>
```

Using a set of values:

```
<constraint>
  <values>
    <value>10/value>
    <value>50</value>
    <value>100</value>
  </values>
</constraint>
```

Example for a string or stringArray element

Using a pattern for validation:

```
<constraint pattern="[A-Za-z]@company.com"/>
```

Using a set of values:

```
<constraint>
  <values>
    <value>blue/value>
    <value>red</value>
    <value>yellow</value>
  </values>
</constraint>
```

defaultValue

The `defaultValue` element specifies the default value for the EMM service to:

- display in the user interface control for the configuration setting.
- send to the app as the value in a key-value pair if the enterprise administrator makes no change to the user interface control.

Important: Carefully consider the best default value for each control. When enterprise administrators do not need to make any changes on the EMM service admin console, your app will be ready to use without either the enterprise administrator or device user doing any work.

Parent elements

- [boolean](#)
- [date](#)
- [float](#)
- [floatArray](#)
- [integer](#)
- [integerArray](#)
- [string](#)
- [stringArray](#)

Contains these elements

The parent element of the `defaultValue` element determines:

- whether [userVariable](#) or [deviceVariable](#) can be a child element of `defaultValue`
- whether more than one child element is allowed
- the allowed contents of a [value](#) element.

The following table shows the allowed child elements and values for the `defaultValue` element, depending on the parent element:

Parent element of <code>defaultValue</code> element	Allowed child elements of <code>defaultValue</code> element
boolean	A value element containing either <code>true</code> OR <code>false</code>
date	A value element containing a date in ISO-8601 format
float	A value element containing a float
floatArray	One or more value elements, each containing a float
integer	A value element containing an integer
integerArray	One or more value elements, each containing an integer
string	One of the following: <ul style="list-style-type: none">• a value element containing a string literal• a userVariable element• a deviceVariable element <p>Note: The <code>defaultValue</code> is not localizable.</p>
stringArray	One or more of the following: <ul style="list-style-type: none">• a value element containing a string literal• a userVariable element• a deviceVariable element <p>Note: The <code>defaultValue</code> is not localizable.</p>

For the list of available user and device variables, see:

- [“EMM service user variables” on page 53](#)

- [“EMM service device variables” on page 54](#)

Examples for a string element

Using at an EMM service user variable:

```
<defaultValue>  
  <userVariable value="emailAddress"/>  
</defaultValue>
```

Using a literal value:

```
<defaultValue>  
  <value>blue</value>  
</defaultValue>
```

Example for a stringArray element

```
<defaultValue>  
  <value>red</value>  
  <value>blue</value>  
  <value>yellow</value>  
</defaultValue>
```

Example for an integer element

```
<defaultValue>  
  <value>red</value>  
</defaultValue>
```

Example for a date element

```
<defaultValue>  
  <value>2015-07-25T22:12:52+00:00</value>  
</defaultValue>
```

deviceVariable

The deviceVariable element specifies the name of an EMM service device variable.

For descriptions and examples of available variables, see [“EMM service device variables” on page 54](#).

Parent element

[defaultValue](#)

Attribute

Attribute	Required or Optional	Description
value	Required	The name of the EMM device variable. Possible values are: <ul style="list-style-type: none">• iccid• imei• imsi• meid• model• phoneNumber• serialNumber• udid• wifiMacAddress

Example

```
<deviceVariable value="IMEI"/>
```

userVariable

The userVariable element specifies the name of an EMM service user variable.

For descriptions and examples of available variables, see ["EMM service user variables" on page 53](#).

Parent element

[defaultValue](#)

Attribute

Attribute	Required or Optional	Description
value	Required	The name of the EMM user variable. Possible values are: <ul style="list-style-type: none">• cn• displayName• dn• emailAddressDomain• emailAddressLocalPart• emailAddress• firstName• lastName• locale• ou• sAMAccountName• userName• upn

Example

```
<userVariable value="emailAddress"/>
```

value

The value element specifies a literal value. The meaning of the value element depends on its parent element:

Parent element of value element	Meaning of child value element
defaultValue element	The default value of a configuration setting.
values element	A valid value of a configuration setting. The parent of the values element is the constraint element.

Parent elements

- [defaultValue](#)
- [values](#)

Example

```
<value>98.6</value>
```

values

The values element defines a set of [value](#) elements, which define the set of valid values of a [float](#), [floatArray](#), [integer](#), [integerArray](#), [string](#), or [stringArray](#) element.

Use the values element when you want to give the enterprise administrator a list of valid values to choose from.

This element is not allowed when the parent element of the [constraint](#) element is [boolean](#) or [date](#).

Parent element

[constraint](#)

Contains these elements

Element	Occurrences	Description
value	Required (1 or more)	A valid value of this configuration setting.

Example

```
<values>
  <value>0</value>
  <value>5</value>
  <value>10</value>
  <value>60</value>
</values>
```

Presentation-related elements

Presentation-related elements define how you want the EMM service to display the configuration settings on the EMM service admin console.

```
presentation (0-1)           Attribute: defaultLocale
  fieldGroup (0+)
    name (1)
      language (1+)         Attribute: value
    field (1+)
  field (0+)                 Attributes: keyName, type
  label (0-1)
    language (1+)          Attribute: value
  description (0-1)
    language (1+)          Attribute: value
  options (0-1)
    option (1+)            Attributes: selected, value
      language (1+)        Attribute: value
```

The presentation-related elements are listed below in alphabetical order.

description

The description element specifies the description of the UI control. The EMM service admin console displays the description with the UI control. Provide information to help the enterprise administrator correctly configure the UI control.

Parent element

field

Contains these elements

Element	Occurrences	Description
language	Required (1 or more)	A localized string to be used as the description for a UI control to be displayed on the EMM service admin console. For more information about which language element the EMM service chooses, see “Localized strings for the admin console” on page 14 .

Example

```
<description>
  <language value="en_US">Automatically populate the username when authentication is required</language>
  <language value="es">Rellenar automáticamente el nombre de usuario cuando se requiere Autenticación</language>
  <language value="fr">Remplir automatiquement le nom d'utilisateur lorsque authentication est nécessaire</language>
</description>
```

field

The field element defines the admin console UI control for a configuration setting (key element such as [string](#), [stringArray](#), [integer](#), and so on).

Note: If a key element has no corresponding [field](#) element, the EMM service creates a default UI control based on the key element. For example, the EMM service can use the `keyName` attribute of the key element as the label for the control.

Parent elements

- [presentation](#)
- [fieldGroup](#)

Attributes

Attribute	Required or optional	Description																
keyName	Required	<p>The keyName attribute of the key element that this UI control represents.</p> <p>Example: keyName="zipcode"</p>																
type	Required	<p>The type of UI control to display on the EMM service admin console.</p> <p>Possible values are:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>checkbox</td> <td>A control, such as radio button or checkbox, for selecting or not selecting one item.</td> </tr> <tr> <td>datetime</td> <td>A control, such as a calendar control, for selecting a date and time.</td> </tr> <tr> <td>hidden</td> <td>No UI control. Use this value to specify a static value that cannot be changed by the enterprise administrator. Be sure to specify a defaultValue in the corresponding key element.</td> </tr> <tr> <td>input</td> <td>A text input control, such as a text box.</td> </tr> <tr> <td>list</td> <td>A control for entering an array of values, such as a list of text boxes. The value that the app receives for such a key is an array of strings.</td> </tr> <tr> <td>multiselect</td> <td>A control, such as a dropdown or combo box, for selecting one or more of many items.</td> </tr> <tr> <td>select</td> <td>A control, such as a dropdown or combo box, for selecting one of many items.</td> </tr> </tbody> </table> <p>Example: type="input"</p>	Value	Description	checkbox	A control, such as radio button or checkbox, for selecting or not selecting one item.	datetime	A control, such as a calendar control, for selecting a date and time.	hidden	No UI control. Use this value to specify a static value that cannot be changed by the enterprise administrator. Be sure to specify a defaultValue in the corresponding key element.	input	A text input control, such as a text box.	list	A control for entering an array of values, such as a list of text boxes. The value that the app receives for such a key is an array of strings.	multiselect	A control, such as a dropdown or combo box, for selecting one or more of many items.	select	A control, such as a dropdown or combo box, for selecting one of many items.
Value	Description																	
checkbox	A control, such as radio button or checkbox, for selecting or not selecting one item.																	
datetime	A control, such as a calendar control, for selecting a date and time.																	
hidden	No UI control. Use this value to specify a static value that cannot be changed by the enterprise administrator. Be sure to specify a defaultValue in the corresponding key element.																	
input	A text input control, such as a text box.																	
list	A control for entering an array of values, such as a list of text boxes. The value that the app receives for such a key is an array of strings.																	
multiselect	A control, such as a dropdown or combo box, for selecting one or more of many items.																	
select	A control, such as a dropdown or combo box, for selecting one of many items.																	

Contains these elements

Element	Occurrences	Description
label	Optional (0-1)	The label of the UI control to be displayed on the EMM service admin console. If a field element does not contain a label element, the EMM service uses the keyName attribute of the corresponding key element.

Element	Occurrences	Description
description	Optional (0 - 1)	The description of the UI control to be displayed on the EMM service admin console.
options	Optional (0 - 1)	A list of option elements, each of which specify localized option names for an option in a <code>select</code> or <code>multiselect</code> UI control. If no options element is included in the field element, the EMM service determines how to display a default <code>select</code> or <code>multiselect</code> UI control. For example, the control can display the values of the values element listed in the constraint element for the corresponding key element.

fieldGroup

The `fieldGroup` element defines a group of configuration settings (key elements such as [string](#), [stringArray](#), [integer](#), and so on) that are to be related on the admin console. The admin console displays a group box control for the user interface fields corresponding to the specified key elements.

Parent element

[presentation](#)

Contains these elements

Element	Occurrences	Description
name	Required (1)	The name of the group box (field group) to be displayed on the EMM service admin console.
field	Required (1 or more)	Defines the admin console UI control for a configuration setting (key element) in the field group.

Example

```
<fieldGroup>
  <!-- The localized name of the field group -->
  <name>
    <language value="en_US">User Information</language>
    <language value="es">Informacion del usuario</language>
    <language value="de">Benutzerinformationen</language>
  </name>

  <!-- Field elements of the fields in the group go here. -->

</fieldGroup>
```

label

The `label` element specifies the label of the UI control to be displayed on the EMM service admin console. If a `field` element does not contain a `label` element, the EMM service uses the `keyName` attribute of the corresponding key element.

Parent element

field

Contains these elements

Element	Occurrences	Description
language	Required (1 or more)	A localized string to be used as the label for a UI control to be displayed on the EMM service admin console. For more information about which language element the EMM service chooses, see "Localized strings for the admin console" on page 14 .

Example

```
<label>  
  <language value="en_US">Login</language>  
  <language value="es">Iniciar Sesión</language>  
  <language value="fr">S'identifier</language>  
</label>
```

language

The language element specifies a localized string to be used as the label, description, option, or name for a UI control or group box to be displayed on the EMM service admin console.

For more information about which language element the EMM service chooses, see ["Localized strings for the admin console" on page 14](#).

Parent elements

- [label](#)
- [description](#)
- [option](#)
- [name](#)

Attribute

Attribute	Description
value	The language code or language-region code for the language element. Example: value="en-US"

Example

```
<language value="es">Iniciar Sesión</language>
```

name

The name element specifies the title of the group box (field group) to be displayed on the EMM service admin console.

Parent element

[fieldGroup](#)

Contains these elements

Element	Occurrences	Description
language	Required (1 or more)	A localized string to be used as the title for the group box (field group) to be displayed on the EMM service admin console. For more information about which language element the EMM service chooses, see " Localized strings for the admin console " on page 14 .

Example

```
<name>  
  <language value="en_US">User Information</language>  
  <language value="es">Informacion del usuario</language>  
  <language value="de">Benutzerinformationen</language>  
</name>
```

option

The option element specifies the localized option names for an option in a `select` or `multiselect` UI control.

If no option element is included for a possible value, the EMM service determines how to display the option in the `select` or `multiselect` UI control. For example, the EMM service can display the value for the option. The value is specified in the [values](#) elements of the [constraint](#) element for the key element corresponding to the field.

Parent element

[options](#)

Attributes

Attribute	Required or optional	Description
selected	Optional	true if this UI control should be initially selected. Otherwise false, which is the default. Example: selected="true"
value	Required	The value that this option element represents. The value must be one of the values in constraint element of the key element corresponding to this field element. Example: value="100"

Contains these elements

Element	Occurrences	Description
language	Required (1 or more)	A localized string to be used as the option name in a UI control to be displayed on the EMM service admin console. For more information about which language element the EMM service chooses, see " Localized strings for the admin console " on page 14 .

Example

```
<option selected="true" value="#FF0000">  
  <language value="en_US">Red</language>  
  <language value="es">Rojo</language>  
  <language value="fr">Rouge</language>  
</option>
```

options

The options element specifies a list of [option](#) elements, each of which specify localized option names for an option in a `select` or `multiselect` UI control.

If no options element is included in a [field](#) element for which the type attribute is `select` or `multiselect`, the EMM service determines how to display the possible values in the `select` or `multiselect` UI control. For example, the EMM service can display the values of the [values](#) element listed in the [constraint](#) element for the corresponding key element.

Parent element

[field](#)

Contains these elements

Element	Occurrences	Description
option	Required (1 or more)	Specifies the localized option names for an option in a <code>select</code> UI control. If no option element is included for a possible value, the EMM service determines how to display the option in the <code>select</code> UI control. For example, the EMM service can display the value for the option. The values are specified in the values elements of the constraint element for the key element corresponding to the field.

Example

```
<options>
  <option selected="true" value="#FF0000">
    <language value="en_US">Red</language>
    <language value="es">Rojo</language>
    <language value="fr">Rouge</language>
  </option>

  <option value="#0000FF">
    <language value="en_US">Blue</language>
    <language value="es">Azul</language>
    <language value="fr">Bleu</language>
  </option>
</options>
```

presentation

The `presentation` element defines how you want the EMM service to display the configuration settings on the EMM service admin console.

If you do not provide a `presentation` element, the EMM service uses information from each key element (such as [string](#), [stringArray](#), [integer](#), and so on) to determine how to create a corresponding UI control.

Parent element

[managedApplicationConfiguration](#) (the root element)

Attribute

Attribute	Required or optional	Description
defaultLocale	Required	<p>Specifies which language element to use for localization if the label, description, option, or name element does not contain a language element for the primary language. The primary language depends on the implementation of the EMM service. It can be, for example, the language that the browser running the admin console is using.</p> <p>Note: Make sure you provide language elements with <code>value</code> attributes that <i>exactly</i> match the <code>defaultLocale</code> attribute.</p> <p>Example:</p> <pre>defaultLocale="en-US"</pre> <p>For more information, see "Localized strings for the admin console" on page 14.</p>

Contains these elements

Element	Occurrences	Description
field	Optional (0 or more)	<p>Defines the admin console UI control for a configuration setting (key element such as string, stringArray, integer, and so on).</p> <p>Note: If no field element is provided for a key element, the EMM service creates a default UI control based on the key element.</p>
fieldGroup	Optional (0 or more)	<p>Defines a group of configuration settings (key elements) that are to be related on the admin console. The admin console displays a group box control for the user interface fields corresponding to the specified key elements.</p>

Example

This example assumes only one configuration setting for a key called `startTime`.

```
<presentation defaultLocale="en-US">
  <field keyName="startTime" type="input">
    <label>
      <language value="en-US">Start Time</language>
      <language value="es-ES">Hora de Inicio</language>
    </label>
    <description>
      <language value="en-US">The time of day to collect data</language>
      <language value="en-US">La hora del día para recoger los datos</language>
    </description>
  </field>
</presentation>
```

EMM service user variables

You can use the EMM service user variables as the default value for your key-value pairs.

See ["Default values for configuration settings" on page 13.](#)

The following table shows the available EMM service user variables:

XML name	Description	Sample of substituted value	Active Directory (AD) or LDAP dependency
cn	Common Name (CN) attribute extracted from the distinguished name	Jane Doe	AD or LDAP
displayName	Display name	Jane Doe, CEO	
dn	Distinguished Name	CN=Jane Doe, OU=NA, OU=Users, OU=XY, DC=myCompany, DC=com	AD or LDAP
emailAddress	Email address	jdoe@myCompany.com	
emailAddressDomain	The domain part of the email address (part after '@')	myCompany.com	
emailAddressLocalPart	The local part of the email address (part before '@')	jdoe	
firstName	First name	Jane	
lastName	Last name	Doe	
locale	Locale	en_US	
ou	Organizational Unit (OU) attribute extracted from the distinguished name	XY	AD or LDAP
sAMAccountName	The Microsoft sAMAccountName attribute	jdoe	AD
username	Login ID (email address format)	jdoe@myCompany.com	
upn	The Microsoft userPrincipalName attribute	jdoe@myCompany.com	AD

EMM service device variables

You can use the EMM service device variables as the default value for your key-value pairs.

See ["Default values for configuration settings" on page 13.](#)

The following table shows the available EMM service device variables:

XML name	Description	Sample of substituted value
iccid	Integrated Circuit Card Identifier	89014104254287052057
imei	International Mobile Equipment Identity	01 342300 291808 3
imsi	International Mobile Subscriber Identity	310150123456789
meid	Mobile Equipment Identifier	A0123456789012
model	Model	iPhone 6
phoneNumber	Phone number	888-555-1212
serialNumber	Serial number	DNRJVL7DTTN
udid	iOS Unique Device Identifier	c752e7052fe5e5ca8166e408c4b48573b5b5bd82
wifiMacAddress	Wi-Fi MAC Address	30:f7:c5:87:e8:78

Managed app configuration XML document example

```
<?xml version="1.0" encoding="UTF-8"?>
<managedAppConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="/com.mobileiron.onetouch/appconfig/appconfig.xsd">
  <version>123</version>
  <bundleId>com.myCompany.myApp</bundleId>
  <dict>
    <boolean keyName="boolkey1">
      <defaultValue>
        <value>true</value>
      </defaultValue>
    </boolean>
    <integer keyName="intkey1">
      <defaultValue>
        <value>20</value>
      </defaultValue>
      <constraint min="1" max="100" nullable="true"/>
    </integer>
    <integer keyName="intkey2">
      <defaultValue>
        <value>25</value>
      </defaultValue>
      <constraint>
        <values>
          <value>15</value>
          <value>20</value>
          <value>25</value>
          <value>30</value>
        </values>
      </constraint>
    </integer>
    <float keyName="floatkey1">
      <defaultValue>
        <value>1.50</value>
      </defaultValue>
      <constraint min="1" max="2"/>
    </float>
    <float keyName="floatkey2">
      <defaultValue>
        <value>20.5</value>
      </defaultValue>
      <constraint>
        <values>
          <value>20.0</value>
          <value>20.1</value>
          <value>20.2</value>
          <value>20.3</value>
          <value>20.4</value>
          <value>20.5</value>
        </values>
      </constraint>
    </float>
    <string keyName="strkey1">
      <defaultValue>
        <userVariable value="cn"/>
      </defaultValue>
      <constraint nullable="true" pattern="CN=.*"/>
    </string>
    <string keyName="strkey2">
      <defaultValue>
        <value>blue</value>
      </defaultValue>
      <constraint>
        <values>
          <value>blue</value>
        </values>
      </constraint>
    </string>
  </dict>
</managedAppConfiguration>
```

```

                <value>red</value>
                <value>green</value>
            </values>
        </constraint>
    </string>
    <integerArray keyName="intarraykey1">
        <defaultValue>
            <value>2048</value>
        </defaultValue>
        <constraint>
            <values>
                <value>1024</value>
                <value>2048</value>
                <value>4096</value>
            </values>
        </constraint>
    </integerArray>
    <floatArray keyName="floatarraykey1">
        <defaultValue>
            <value>1.5</value>
        </defaultValue>
        <constraint>
            <values>
                <value>1.2</value>
                <value>1.3</value>
                <value>1.4</value>
                <value>1.5</value>
            </values>
        </constraint>
    </floatArray>
    <stringArray keyName="strarraykey1">
        <defaultValue>
            <userVariable value="emailAddress"/>
        </defaultValue>
        <constraint pattern="[A-Za-z]@company.com"/>
    </stringArray>
    <date keyName="datekey1">
        <defaultValue>
            <value>2015-07-21T16:29:30Z</value>
        </defaultValue>
    </date>
    <integer keyName="hiddenkey1">
        <defaultValue>
            <value>1</value>
        </defaultValue>
    </integer>
</dict>
<presentation defaultLocale="en-US">
    <fieldGroup>
        <name>
            <language value="en-US">Critical Settings</language>
        </name>
        <field keyName="boolkey1" type="checkbox">
            <label>
                <language value="en-US">Enabled</language>
            </label>
            <description>
                <language value="en-US">Click this checkbox to enable.</language>
            </description>
        </field>
        <field keyName="intkey2" type="select">
            <label>
                <language value="en-US">Connection Timeout</language>
            </label>
            <description>
                <language value="en-US">The connection timeout in seconds.</language>
            </description>
        </field>
    </fieldGroup>
</presentation>

```



```

    </description>
    <options>
      <option value="15">
        <language value="en-US">15</language>
      </option>
      <option value="20">
        <language value="en-US">20</language>
      </option>
      <option value="25">
        <language value="en-US">25</language>
      </option>
      <option value="30">
        <language value="en-US">30</language>
      </option>
    </options>
  </field>
</fieldGroup>
<field keyName="strarraykey1" type="input">
  <label>
    <language value="en-US">Email Address</language>
  </label>
  <description>
    <language value="en-US">Please enter the user's email address.</language>
  </description>
</field>
<field keyName="datekey1" type="datetime">
  <label>
    <language value="en-US">Start Date</language>
  </label>
  <description>
    <language value="en-US">Please enter a start date.</language>
  </description>
</field>
  <field keyName="hiddenkey1" type="hidden"/>
</presentation>
</managedAppConfiguration>

```

plist format sent to device

The EMM service sends the key-value pairs to the device in a plist. iOS puts the key-value pairs from the plist into NSUserDefaults. Your app does not access the plist directly. It is included here only as background information.

plist format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>

    <!-- This ManagedAppConfigurationVersion key has the value of the -->
    <!-- version element in the XML document -->

    <key>ManagedAppConfigurationVersion</key>
    <integer>2</integer>

    <!-- The plist then has a key-value pair for each configuration setting. -->
    <!-- The name of the key corresponds to the keyName attribute of a key element -->
    <!-- in the XML document. -->
  </dict>
</plist>
```

plist example with each type of key

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>ManagedAppConfigurationVersion</key>
    <string>2</string>

    <key>boolean_key_name</key>
    <true/>

    <key>date_key_name</key>
    <date>2015-07-25T21:19:30Z</date>

    <key>float_key_name</key>
    <real>2.99</real>

    <key>float_array_key_name</key>
    <array>
      <real>2.99</real>
      <real>9.99</real>
      <real>19.99</real>
    </array>

    <key>integer_key_name</key>
    <integer>30</integer>

    <key>integer_array_key_name</key>
    <array>
      <integer>0</integer>
      <integer>50</integer>
      <integer>100</integer>
    </array>

    <key>string_key_name</key>
    <string>Lou Brock</string>

    <key>string_array_key_name</key>
    <array>
      <string>St. Louis Cardinals</string>
      <string>San Francisco Giants</string>
      <string>Los Angeles Dodgers</string>
    </array>
  </dict>
</plist>
```